Understanding student input for tutorial dialogue in procedural domains

Understanding student input for tutorial dialogue in procedural domains

Myroslava O. Dzikovska*, **Charles B. Callaway***, **Matthew Stone**†,*, **Johanna D. Moore***
*Human Communication Research Centre, University of Edinburgh,
Edinburgh, EH8 9LW, United Kingdom
{mdzikovs,ccallawa,jmoore}@inf.ed.ac.uk
† Department of Computer Science, Rutgers University, Picataway, NJ, 08845-8020
mdstone@cs.rutgers.edu

Abstract

We present an analysis of student language input in a corpus of tutoring dialogue in the domain of symbolic differentiation. Our focus on procedural tutoring makes the dialogue comparable to collaborative problem-solving (CPS). Existing CPS models describe the process of negotiating plans and goals, which also fits procedural tutoring. However, we provide a classification of student utterances and corpus annotation which shows that approximately 28% of non-trivial student language in this corpus is not accounted for by existing models, and addresses other functions, such as evaluating past actions or correcting mistakes. Our analysis can be used as a foundation for improving models of tutoring dialogue.

1 Introduction

In domains from mathematics to maintenance, human tutors often instruct students by coaching them through procedures they must learn. There seems to be a natural analogy between this activity and people's collaborative problem solving (CPS) when they jointly pursue real-world goals. For example, in both cases, interlocutors must talk through what they have accomplished so far, what to do next, and how to do it. The analogy has prompted tutoring researchers such as Rickel et al. (2001) to model procedural tutoring in terms of existing models of CPS.

This research program promises to enrich tutorial dialogue systems by leveraging a rich research tradition (Grosz and Sidner, 1986; Grosz and Kraus, 1996; Lochbaum, 1998; Blaylock and Allen, 2005). But it assumes that students and tutors use the same kinds of utterances to make the same kinds of moves as found in previously studied collaborative domains. In this paper, we offer an empirical assessment of this assumption.

We report an analysis of a corpus of dialogues for tutoring a mathematics procedure, symbolic differentiation, which has been annotated reliably for a coarse categorization of student behavior. Our analysis suggests that students and tutors work together to maintain a shared understanding of the procedure being carried out and learned, so in broad terms, the CPS model of procedural tutoring is a good one. However, we find that about 28% of student utterances are not covered explicitly by existing models of procedural tutoring and CPS. CPS research seems to have overlooked much of the talk that interlocutors use to reach common ground not just about what they are *going to do* but about what they *have done*. Such moves may be particularly frequent in procedural tutoring because it aims at student understanding, and must accommodate the errors students make while learning.

Our work is informed by the long-term goal of building interactive natural language dialogue systems that reproduce the increased effectiveness of one-on-one human tutoring over classroom instruction (Bloom, 1984). Crucially, from this perspective, the new moves we identify not only occur frequently, but also seem to offer useful information about what the student has learned and what feedback they need — information that would not be available just from student actions or even moves in earlier CPS models. So we advance the development of richer formal models as a challenge for future theoretical and applied research in tutorial dialogue.

We begin by describing in Section 2 our corpus and the domain model. In Section 3 we discuss our annotation scheme, how it relates to the Shared Plan model (Grosz and Sidner, 1986; Grosz and Kraus, 1996) as well as to specific implementations of CPS for tutoring, particularly that of Rickel et al. (2001), and the role different student inputs play in tutoring. In Section 4 we discuss the corpus annotation and analysis, concluding with related and future work in Section 5.

2 Corpus and Tutoring System

2.1 Corpus Collection

The task domain we used in this study is tutoring symbolic differentiation. The task involves applying a set of differentiation rules (the chain rule, the product rule, the sine rule etc.), and the algebraic transformations to bring the result to a normal form. We analyzed a corpus of interactions between students and experienced tutors in this domain, which we are using to study tutorial dialogue and to build a tutoring system.

The data consists of 19 transcripts of 30-minute tutoring sessions conducted via a chat interface. During the session, the tutor gave the student a sequence of problems on using the chain rule until they ran out of time. The student did not propose or choose the problems.[1] Two experienced mathematics instructors (as tutors) and 14 first-year mathematics or science undergraduate students who were learning differentiation in a calculus course at the University of Edinburgh were paid to participate.

The data collection environment separated students from tutors physically. They could only exchange utterances via a chat interface where interlocutors could send each other text messages. Complex mathematical expressions could be entered using a special editor, and text and formulas could be intermixed. The tutor could observe the student's actions in real-time on a second screen. Students and tutors were trained to use the interfaces prior to the data collection session.

The resulting corpus consists of 19 dialogues (5 students returned twice) and contains 1650 utterances (with textual and symbolic parts), 5447 words and 559 formulas.

[1] This restriction was not explicitly part of the experimental design, but in practice students did not attempt to choose or negotiate on problems to solve.

2.2 Domain Model

Based on the corpus analysis, we built a model of differentiation with the chain rule which instantiates possible actions in our domain. Our model consists of 5 basic steps: 1) rewriting the function to a recognizable form which can be matched against a differentiation rule; 2) differentiating the outer layer; 3) differentiating the inner layer; 4) combining the results by multiplying; and 5) simplifying the resulting expression. To support tutoring, the actions applicable at each step may take the form of either applying a correct differentiation rule, or else using an incorrect (buggy) rule (Brown and Burton, 1978).

This domain model integrates both correctly executed actions and errors as possible steps in problem-solving. However, when a student enters a formula, it may be ambiguous with respect to which of the steps in the domain model it matches, especially in the presence of errors. Moreover, the students do not always know all the necessary steps. For example, they frequently do not realize that the resulting expression after step 4) needs to be simplified. Student language may provide additional important clues to resolve such ambiguities, as discussed in Section 3.

We implemented a prototype system, BEEDIFF (Callaway et al., 2006) with a domain reasoner which supports the task model described above (Zinn, 2006) without language input. The domain reasoner acts as a plan recognizer by fitting submitted formulas with the task model. One of the goals of our corpus analysis is to identify the types of student utterances which occur frequently in our corpus, for example, help requests, and in the subsequent analysis to identify the appropriate pedagogical and dialogue strategies to use in response in our dialogue system.

3 Annotation Scheme

The goal of our corpus analysis is to identify student language in tutorial dialogue consistent with existing CPS models, as well as the language which is not accounted for directly. In our comparison we focus on the Shared Plan model of discourse (SP model) (Grosz and Sidner, 1986) and its implementation in the COLLAGEN PACO tutoring agent (Rickel et al., 2001). PACO implements Lochbaum's (1998) intention recognition algorithm which is based on the SP model. The implementation supports a subset of an artificial

T: Differentiate $sqrt(x^3 - 9x)$
S: $(x^3 - 9x)^{-1}$
is this equal to the question (**help-specific**)
T: No. Remember that $\sqrt{(z)} = z^{1/2}$ Can you rewrite the question now?
S: would that be $(x^3 - 9x)^{1/2}$ (**task-progression**)

Figure 1: A sample dialogue with student utterances containing language with our annotations.

CPS language proposed by Sidner (1994), which gives an account of beliefs and intentions which occur in utterances modeled by the SP model.

Blaylock and Allen (2005) provide another CPS model of task-oriented dialogue. The model is broadly compatible with the SP model, but focuses primarily on interlocutors' collaboration in negotiating tasks and resources amid task execution. It does not directly address how explanations and assessment of completed steps fits into the dialogue, and has not been extended to tutoring, hence our main focus on the SP model and PACO.

The classes of utterances implemented in PACO are agreeing and disagreeing, proposing tasks, actions and goals, indicating that a task has been accomplished, asking about or proposing task and action parameters, asking and proposing how tasks should be accomplished, and asking what to do next. By contrast, our classification differentiates the following kinds of student behavior: help requests; queries about next steps; comments on cognitive state (*i.e.* explaining what they are doing, or stating their knowledge or beliefs related to the problem); progress evaluation; and more general dialogue behavior related to agreement and clarification. Our model is deliberately coarse-grained in order to explore the properties of the dialogue. The categories thus indicate directions for future specification and implementation, rather than moves that could be directly formalized in a detailed model of tutorial dialogue. A fragment of annotated dialogue is shown in Figure 1.

Table 1 presents a correspondence between the language in the PACO model and our classifications, which are discussed in more detail below. As the table shows, the SP model, along with its implementation in PACO, provides ways of thinking about and representing discourse that can describe a wide range of student utterances in tutoring. However, the tutoring setting makes available to students a variety of utterances whose form, content or function differs from those more traditional collaborators might typically use.

Help requests. These are utterances that indicate that the student does not know what the next step is, or does not know how to perform a specific step (or perhaps is not confident enough to perform it). We subdivided help requests into generic and specific requests. Generic requests indicate that the student is stuck, for example, "I don't know what to do", but don't provide further clues as to what the problem is. Specific help requests identify explicitly what the students do not know, *e.g.*, "I don't know what the common factor is", or "$z = cos(x)$. I don't know about y".

Generic help requests correspond to asking what to do next in the CPS model. Specific help requests correspond either to asking about task and action parameters, or to asking about the appropriate recipe (how to perform a given step).

Verifying next steps and goals. Instead of doing the step directly, students often describe it first (without indicating the end result), in order to get confirmation from the tutor, for example, "Should I multiply the 3 and the 15", or "Should I simplify this further?" This utterance category corresponds to proposing steps in the CPS model. However, the language is quite different compared to that typically seen in CPS, where proposals are normally offered up for negotiation, *e.g.*, "Let's start engine two". This shows that students are not really negotiating how to select the next step from among a set of possibilities. Rather, students who produce this type of utterance may be uncertain how to proceed, and need the tutor's help.

Clarification requests. As in any dialogue, there are cases where students do not understand what the tutor has said (as opposed to not knowing what to do next, or how to perform a step). Students then attempt to clarify the meaning of the comment, either on the level of terminology, or a more conceptual level. For example, "What do you mean by $3x^2$" or "How can there be an inner layer when there is no power". Clarifications are part of conversational expertise, and should be accounted for by general dialogue models.

Dialogue progression. These are steps students usually take to acknowledge the tutor's instructions, to indicate that they understand the mate-

Student utterance class	PACO language type
Generic help request	asking what to do next
Specific help requests	asking about task and action parameters, or how tasks should be accomplished
Next step verification	Proposing steps, goals, action parameters or recipes
Dialogue progression	Agreeing and disagreeing
Progress evaluation	Consistent with the SP model, but not explicitly in PACO
Explanations	Indicating which task was accomplished, other kinds not modeled in detail
Task progression	Consistent with the SP model, but not explicitly in PACO
Stating their knowledge	Consistent with the SP model, but not explicitly in PACO
Editing the solution in language	Can be treated as just performing the step directly

Table 1: Correspondences between our coding scheme and language categories in (Rickel et al., 2001)

rial,[2] and in general to establish that the tutor has been understood and to advance the dialogue. We expect that these dialogue moves can be covered by a general model of collaborative dialogue, because they are not in any way specific to tutoring.

Progress evaluation. Students often either evaluate their own progress (*e.g.,* "I made a mistake"), or ask the tutor to evaluate their progress (*e.g.,* "Is this right?").[3] These utterances are consistent with the SP model, but in PACO only tutors' evaluations of students' progress are included, and they are treated as accepting or rejecting the student actions. This is not appropriate for students evaluating their own progress, or asking for evaluation.

One may think progress evaluation is an example of grounding (Clark and Brennan, 1991), rather than a specific CPS move. However many other CPS moves actually let interlocutors show or check that they have achieved mutual understanding. Moreover, modeling progress evaluation is not just a matter of allowing discussion at set points in plan-execution. Progress evaluation suggests that in tutoring, unlike domain-oriented problem solving, tutors allow students to make mistakes and expect that they may not necessarily recognize problems. The whole CPS process for tutoring therefore has to underconstrain actions to include errors and underconstrain context to allow for ignorance, and has to be more explicit about how progress is evaluated, including allowing students to ask for evaluation or give it themselves.

Explanations. Students may verbalize what they are doing as they are solving the problem ("forward-looking" explanations) or, usually after a mistake, reflect on what they did and why ("backward-looking" explanations). An example forward-looking explanation is the utterance "and put back in the original form $15x^2/3(5x^3-6)^4$". In this case, the formula in the utterance is the re-writing of the previous solution step, which is necessary to finish up the differentiation procedure.[4] An example of a backward-looking explanation is, in reply to the tutor's request "Why did you do that?", "because I thought you multiplied the powers when they were bracketed". Tutors occasionally asked the students how they arrived at their (incorrect) solutions, but sometimes students offered their own explanations spontaneously after the tutor corrected their mistake. Explanations of student actions have also been observed in tutoring algebra proofs (Wolska and Kruijff-Korbayová, 2004).

These explanations are not modeled in sufficient detail by existing CPS models. Sidner's negotiation language contains `provide-support` as a possible action, and the PACO implementation incorporates the tutor giving explanations within recipes specific to tutoring. A student's asking of "why" questions is modeled as a proposal to the tutor to provide support for what should be done. These models only describe utterances that motivate actions *before* they are agreed on, while our explanations often accompany actions as they are

[2] Saying "I don't understand" would usually be classified as a help request unless it is related to surface form of tutor's words

[3] We classified as progress evaluation only the items which were "content-free" and could be covered with generic "oops" and "am I right?" buttons. If the student described the problem in more detail, the utterance was classified as an edit or an explanation.

[4] If the language was not present, the formula by itself would be counted as performing the re-writing step.

done or afterward.

Explanations should be modeled more explicitly as part of the tutoring process. Forward-looking explanations can be seen as behaviors that disambiguate the place of an ongoing action in the plan. While typically collaborators agree on actions before they do them, in some cases one collaborator may decide to act independently to further the joint activity. In such cases, the actor may have to describe the action for their collaborators to recognize the step they are performing. We are not surprised to see this more prominently in tutoring than in other CPS because students are being coached to carry out the procedure on their own.

Backward-looking explanations occur after the relevant problem-solving step has been completed. We can see why such explanations might further students' and tutors' joint activity. These moves may allow students to provide evidence about their understanding of the rules and relationships involved in problem-solving, although a second possible motive is social (to allow students to save face). Such moves may therefore contribute to patterns of interaction between tutor and student that establish a correct mutual understanding of the subject-matter that students should learn. Again, it is no surprise that such backward-looking activity might be more frequent in learning dialogues than typical CPS dialogues, which simply aim at achieving real-world goals.

Task progress indications. Students often indicate whether they are continuing with the problem, or are finished, with progress markers like "first", or "the final answer is ...". These again are consistent with the general SP model where they correspond to cue phrases starting a new discourse segment, under the assumption that each student action starts a new discourse segment which can end immediately when a tutor accepts it, or continue with remediation. Current implementations, however, do not reason about these cues specifically. Similarly to forward-looking explanations, domain reasoning should be sufficient to infer where the student is in executing the task without these markers. However, when the student provides them it may be important for tutoring, because they explicitly indicate where the student thinks she is in the process of solving the problem. Consider the case when the student is differentiating a function and writes $-3 * sin(x)^{-3}$. This expression should be simplified, and it is not always clear if the student thinks that he is done, in which case the tutor needs to remind them to simplify the formula, or if the student is still working on the problem and will simplify on the next step. But if the student says "the final answer is $-3 * sin(x)^{-3}$", then it is a clear indication that he thinks the problem is finished, and the tutor needs to intervene.

Stating knowledge of rules and principles. Students make statements (correct or incorrect) about the rules or principles they know, for example "The derivative of $\sin$ is $\cos$". These usually don't appear by themselves, but are used to support general meta-level tutoring talk, in particular help requests and explanations. We chose to tag these utterances as a separate class because they contain very explicit statements about what students know and believe, as opposed to more indirect indications when students state what they don't know in a help request. In the general SP framework (not explicitly covered in the PACO model) this corresponds to stating or proposing recipes. However, in CPS, proposing recipes is done at the negotiation stage, where different courses of actions are possible to achieve a goal. In a tutorial setting, the function of stating rules is to support tutoring rather than problem-solving per se — it is an attempt from the student to expose their knowledge to the tutor, with the goal of confirming it is complete and correct.

Using language to edit answers. Students sometimes describe a portion of the answer instead of providing the full formula, for example, "Ah, so the top part is $-15x^3$", or correct themselves immediately after supplying the answer (without tutor intervention), *e.g.,* "I meant to put a to the power 6 on the bracket". These utterances can be viewed as doing the step directly in the CPS model in most cases, equivalent to submitting a full formula. Some of these are specific to mathematical dialogue, where the mixture of informal and formal language can be used to describe math expressions (Wolska and Kruijff-Korbayová, 2004). However, self-corrections can be important in other domains, especially if student actions are "non-reversible" (*e.g.,* pressing a button in a simulator). These cases may then require a different strategy on the part of the tutor.

Input not related to tutoring or problem-solving. Task management (e.g, transitions between problems, greetings and closings), and jokes are obviously part of social interaction in any conversation. We expect these to be accounted for with a more general model of conversation, and tagged them as a class of “non-task-related”.

4 Corpus Analysis and Discussion

We annotated student input in 19 dialogues with this scheme. There were a total of 656 student utterances. Out of those, 323 (49%) contained only a mathematical formula contributing to the solution and no language. Our annotation was done over the remaining 333 student utterances which contained at least one word. Out of those, 99 (30%) were judged as not relevant to tutoring or the task of differentiation (greetings and closings, transitions between tasks, jokes, etc.). The distribution of tags among the 234 remaining utterances in our corpus is shown in Table 2. To verify the inter-rater reliability, two annotators independently coded four dialogues (102 utterances) with the scheme, resulting in inter-rater agreement of 84% and $\kappa = 0.78$ (*i.e.* ’good’ agreement).

The categories of student input which are not directly accounted for in the CPS model (evaluation, explanation and knowledge) together account for about 28% of all student language input. This underscores the importance of including those phenomena in a model of tutoring dialogue. Our categorization is a first step in identifying the phenomena which need to be accounted for in a formal model of tutorial dialogue in procedural domains, which is the next step in our work.

Other questions which arise in this line of research are the importance of the individual categories from the point of view of practical systems, as well as the importance of student language in general in tutoring. Our study contributes to answering these questions.

Many student utterances offer information for student modeling that goes beyond what can be derived from the sequence of steps the students execute. In particular, specific help requests, explanations and knowledge statements give indications of student knowledge and misconceptions, and task progression markers and evaluations may help evaluate student knowledge as well as their confidence level.

These categories cover 38% of student utterances, which can be interpreted as an indicator that dialogue participants considered it important in some way. This is not sufficient to make definitive conclusions about the importance of language in tutoring dialogue, because we do not know if the students who used more language learned more.[5] However, our data contain tutors’ assessments of student aptitude, and we plan on investigating if they correlate with the use of language.

Tag	Count	Tag %	Mean	Stdev
Help requests	47	21%		
generic	27	12%	1.42	1.64
specific	20	9%	1.05	1.31
Step requests	19	8%	1.00	1.41
Clarifications	5	2%	0.26	0.56
Dialogue progression	54	23%	2.84	2.14
Edit	15	7%		
forward	9	4%	0.47	0.84
backward	6	3%	0.32	0.58
Evaluation	31	13%		
request	17	7%	0.89	1.20
state	14	6%	0.74	0.87
Explanation	17	8%		
forward	4	2%	0.21	0.42
backward	13	6%	0.68	1.25
Knowledge	15	6%		
global	10	4%	0.53	1.22
problem	5	2%	0.26	0.56
Task progression	5	2%	0.26	0.93
other	26	11%	1.37	2.19

Table 2: Tag distribution in our corpus. Tag % is the percentage of tag occurrences out of the overall tag count; mean and stdev refer to the average number of tag occurrences per dialogue.

Different students used different strategies in their language. For example, the percentage of backward explanations varied from 0 to 27%, and the percentage of specific help requests from 0 to 33%. Thus it is difficult to make predictions comparing the frequencies of individual tags in our corpus. One of the important tasks of further corpus analysis is to determine the cause of this variation, which may be due to individual differences, student aptitude and motivation,[6] or other features

[5] This study did not measure learning gains, which is necessary to assess the amount learned by each student.

[6] We observed that poor students generally talked more and were more specific in their requests, which needs to be confirmed with further analysis

of the interaction.

Most of the categories important to student modeling discussed above rely on non-trivial language which is dependent on the context and cannot be covered by a set of simple questions or buttons (specific help requests, step requests, clarification requests, explanations, and knowledge statements). This correspondence again suggests the importance of extending the models to cover these more complex interactions. The next step would be to determine which tutorial strategies would be appropriate in response to each of those utterance classes, and confirm the correlation with corpus analysis.

5 Related Work

Many tutoring systems for procedural tasks have been built around simulation environments (Rickel et al., 2001; Pon-Barry et al., 2004; Ong and Noneman, 2000). These systems use a task model, augmented with plan recognition, to recognize student actions and intentions and provide feedback and directions accordingly, with very limited student language input. For example the NASA RPOT tutor (Ong and Noneman, 2000) is based on a generic task tutor toolkit which contains a task model and can answer 3 questions: "What do I do", "Why do I do that", and "How do I do that". Rickel et al. (2001) accounts for a subset of student input consistent with the artificial CPS modeling language (Sidner, 1994). Our paper continues this line of work by investigating the student language not covered by existing models.

Shah et al. (2002) provide a model of student initiatives and tutor's responses in the CIRCSIM system (i.e. utterances which go beyond responding to tutor's questions), classifying them along four dimensions. Our classification is closest to their communicative goal dimension, which includes requests for confirmation and for information, challenging the tutor, refusal to answer and conversational repair. Our categorization covers all student utterances regardless of the initiative, with the goal of building model of tutorial dialogue covering the behavior of both dialogue participants, and we intend to study the classes of tutor utterances which are the appropriate follow-ups to student utterances in the future.

A large amount of work in the tutoring literature is dedicated to modeling tutoring strategies, that is, the actions the tutor takes during the interaction (McArthur et al., 1990; Zhou et al., 1999; Pilkington, 1999; Graesser et al., 1999; Pon-Barry et al., 2004). McArthur et al. (1990) propose a model of tutoring in solving algebraic equations in which problems are solved step-by-step according to the task model. Tutors execute "microplans" at each step, which consist of introducing a problem or a step, solving it (done by the student generally), evaluation, remediation if necessary, and an optional wrapup step where the tutor may summarize the step or the problem. This model is "tutor-centric" in the sense that it does not account for the student's actions. Our goal is to develop a model of both student and tutor behavior, which can be used to inform the implementation of a tutorial dialogue system.

Much research has been done in identifying what makes human-human tutoring effective. Self-explanation (Chi et al., 1989), interactivity (VanLehn et al., to appear), and student initiative and "student talk" (Core et al., 2003) have been studied as possible predictors of student learning. But there is currently no definitive study confirming that talking in natural language, and specifically what kind of language, improves learning compared to reading or limited forms of input such as multiple-choice answers.

We are currently conducting a study to evaluate the role of student language input in tutoring. We analyzed a corpus of human-human tutoring dialogues (in a conceptual domain) where the teaching material was designed to elicit a different amount of student language under different conditions. This will allow us to see if there is a correlation between the amount and type of language students use and learning gains. Additionally, we are considering a study in our current domain comparing tutoring with free language input to tutoring where students are only allowed to input formulas and have a small set of buttons to ask for help and confirm or disconfirm their understanding of what the tutor said. We are planning to use these studies to gain further understanding of the role of student natural language input in learning from tutoring.

6 Conclusions

We provided a description of student language in a corpus of procedural tutoring which can serve as an initial model for implementing a tutorial system. We identified student language categories which are not sufficiently modeled in the existing

CPS model, and showed that 28% of student utterance in our corpus fall under those categories. All of those classes fall under the categories of utterances which may be important to student modeling. We argue that the existing CPS models need to be extended to cover these classes of utterances in tutoring dialogue. Our scheme provides an initial categorization of phenomena which need to be included in formal models, as well as working dialogue systems to account for student language in addition to actions.

Acknowledgments

Helen Pain and Kaśka Poryska-Pomsta collected the data used in this analysis. This work is supported by a grant from The Office of Naval Research number N000149910165, European Union 6th framework programme grant EC-FP6-2002-IST-1-507826 (LeActiveMath), NSF HLC 0308121, the Leverhulme Trust and Rutgers University.

References

N. Blaylock and J. Allen. 2005. A collaborative problem-solving model of dialogue. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pages 200–211, Lisbon, September.

B. S. Bloom. 1984. The two sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13:3–16.

J. S. Brown and R. R. Burton. 1978. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science: A Multidisciplinary Journal*, 2(2):155–192.

C. Callaway, M. Dzikovska, C. Matheson, J. Moore, and C. Zinn. 2006. Using dialogue to learn math in the LeActiveMath project. In *Proceedings of the ECAI Workshop on Language-Enhanced Educational Technology*, pages 1–8, August.

M. T. H. Chi, M. Bassok, M. W. Lewis, P. Reimann, and R. Glaser. 1989. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2):145–182.

H. H. Clark and S. E. Brennan. 1991. Grounding in communication. In L. B. Resnick, J. M. Levine, and S. D. Teasley, editors, *Perspectives on Socially Shared Cognition*, pages 127–149. American Psychological Association, Washington, D.C.

M. G. Core, J. D. Moore, and C. Zinn. 2003. The role of initiative in tutorial dialogue. In *Proceedings of EACL-03*, pages 67–74.

A. C. Graesser, P. Wiemer-Hastings, P. Wiemer-Hastings, and R. Kreuz. 1999. Autotutor: A simulation of a human tutor. *Cognitive Systems Research*, 1:35–51.

B. Grosz and S. Kraus. 1996. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357.

B. J. Grosz and C. L. Sidner. 1986. Attentions, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

K. E. Lochbaum. 1998. A collaborative planning model of intentional structure. *Computational Linguistics*, 24(4):525–572.

D. McArthur, C. Stasz, and M. Zmuidzinas. 1990. Tutoring techniques in algebra. *Cognition and Instruction*, 7:197–244.

J. C. Ong and S. R. Noneman. 2000. Intelligent tutoring systems for procedural task training of remote payload operations at NASA. In *I/ITSEC 2000*.

R. M. Pilkington. 1999. Analysing educational discourse: The discount scheme. Technical Report 99/2, Computer Based Learning Unit, Univ. of Leeds.

H. Pon-Barry, B. Clark, K. Schultz, E. O. Bratt, and S. Peters. 2004. Advantages of spoken language interaction in dialogue-based intelligent tutoring systems. In *Proceedings of ITS-2004*, pages 390–400.

J. Rickel, N. Lesh, C. Rich, C. Sidner, and A. Gertner. 2001. Building a bridge between intelligent tutoring and collaborative dialogue systems. In *Proceedings of AIED-2001*, pages 592–594.

F. Shah, M. W. Evens, J. Michael, and A. Rovick. 2002. Classifying student initiatives and tutor responses in human keyboard-to-keyboard tutoring sessions. *Discourse Processes*, 33(1).

C. L. Sidner. 1994. An artificial discourse language for collaborative negotiation. In *Proceedings of AAAI*, pages 814–819.

K. VanLehn, A. C. Graesser, G. T. Jackson, P. Jordan, A. Olney, and C. P. Rosé. (to appear). When are tutorial dialogues more effective than reading? *Cognitive Science*.

M. Wolska and I. Kruijff-Korbayová. 2004. Analysis of mixed natural and symbolic language input in mathematical dialogs. In *Proceedings of ACL-04*, pages 25–32.

Y. Zhou, R. Freedman, M. Glass, J. A. Michael, A. A. Rovick, and M. W. Evens. 1999. What should the tutor do when the student cannot answer a question? In *12th FLAIRS Conference*, pages 187–191.

C. Zinn. 2006. Supporting tutorial feedback to student help requests and errors in symbolic differentiation. In *Proceedings of the International Conference on Intelligent Tutoring Systems*, pages 349–359.